We claim:

1. A method of managing a relational database comprising:

   a. receiving queries in a query language;

   b. representing the queries in accordance with a declarative language paradigm;

   c. converting the queries represented in a declarative language paradigm to an imperative language; and

   d. executing the imperative language queries.

2. The method of claim 1 comprising converting the query language to an intermediate declarative representative, and thereafter converting the query to an imperative language representation of the queries and executing the imperative language queries.

3. The method of claim 1 comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries.

4. The method of claim 1 wherein the declarative language is chosen from the group consisting of ML, LISP, and HASKELL.

5. The method of claim 1 wherein the imperative language is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

6. A method of managing a relational database comprising:

   a. receiving queries in a query language;

   b. converting the query language to an imperative language representation of a declarative language representation of the queries in accordance with a declarative language paradigm; and

   c. executing the imperative language queries.

7. The method of claim 6 wherein the declarative language is chosen from the group consisting of ML, LISP, and HASKELL.

8. The method of claim 6 wherein the imperative language is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

9. A method of managing a relational database comprising:

   a. receiving queries in a query language;

   b. representing the queries in accordance with a declarative language paradigm;

c. converting the queries represented in a declarative language paradigm to a data structure that is effectively interpreted using an imperative language; and

d. executing the imperative language queries.

10. The method of claim 9 comprising converting the query language to an intermediate declarative representative, and thereafter converting the query to a data structure efficiently interpreted using an imperative language representation of the queries and executing the imperative language queries.

11. The method of claim 9 comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries.

12. The method of claim 9 wherein the declarative language is chosen from the group consisting of ML, LISP, and HASKELL.

13. The method of claim 9 wherein the imperative language is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

14. A database management system adapted to process queries in a pervasive computing environment, said pervasive computing environment comprising at least one client adapted to interact with a server over connection services, said at least one client controlled and configured to

a. receive queries in a query language;

b. represent the queries in accordance with a declarative language paradigm;

c. convert the queries represented in a declarative language paradigm to an imperative language; and

d. execute the imperative language queries.

15. The system of claim 14 comprising converting the query language to an intermediate declarative representative, and thereafter converting the query to an imperative language representation of the queries and executing the imperative language queries.

16. The system of claim 14 comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries.

17. The system of claim 14 wherein the declarative language is chosen from the group consisting of ML, LISP, and HASKELL.

18. The system of claim 14 wherein the imperative language is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

19. A database management system adapted to process queries in a pervasive computing environment, said pervasive computing environment comprising at least one client adapted to interact with a server over connection services, said at least one client controlled and configured to

a. receive queries in a query language;

b. represent the queries in accordance with a declarative language paradigm;

c. convert the queries represented in a declarative language paradigm to a data structure that is interpreted using an imperative language; and

d. execute the imperative language queries.

20. The system of claim 19 comprising converting the query language to an intermediate declarative representative, and thereafter converting the query to an imperative language representation of the queries and executing the imperative language queries.

21. The system of claim 19 comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries.

22. The system of claim 19 wherein the declarative language is chosen from the group consisting of ML, LISP, and HASKELL.

23. The system of claim 19 wherein the imperative language is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

24. A program product comprising computer readable program code on one or more media, said program code being capable of controlling and configuring a computer system having one or more computers to perform the process of

a. receiving queries in a query language;

b. representing the queries in accordance with a declarative language paradigm;

c. converting the queries represented in a declarative language paradigm to an imperative language; and

d. executing the imperative language queries.

25. The program product of claim 24 comprising converting the query language to an intermediate declarative representation thereof, and thereafter converting the query to an imperative language representation of the queries and executing the imperative language queries.

26. The program product of claim 24 comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries.

27. The program product of claim 24 wherein the declarative language is chosen from the group consisting of ML, LISP, and HASKELL.

28. The program product of claim 24 wherein the imperative language is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.

29. A program product comprising computer readable program code on one or more media, said program code being capable of controlling and configuring a computer system having one or more computers to perform the process of

a. receiving queries in a query language;

b. representing the queries in accordance with a declarative language paradigm;

c. converting the queries represented in a declarative language paradigm to a data structure that is interpreted by an imperative language; and

d. executing the imperative language queries.

30. The program product of claim 29 comprising converting the query language to an intermediate declarative representative thereof, and thereafter converting the query to an imperative language representation of the queries and executing the imperative language queries.

31. The program product of claim 29 comprising converting the query language to an imperative language representation of the declarative language and executing the imperative language queries.

32. The program product of claim 29 wherein the declarative language is chosen from the group consisting of ML, LISP, and HASKELL.

33. The program product of claim 29 wherein the imperative language is chosen from the group consisting of C, C++, Java, Modula2, and SmallTalk.